27 citations found. Retrieving documents...

Connor R., Brown A.L., Carrick R., Dearie A. & Morrison R. "*The Persistent Abstract Machine*". In "Persistent Object Systems". Workshops in Computing, Rosenberg J. & Koch D. (eds.) pp. 279-288 (Springer-Verlag, 1990.

This paper is cited in the following contexts:

First 50 documents

Cache Coherency and Storage Management in a Persistent Object.. - Koch, al. (1990) (13 citations) (Correct)

....in an environment that is robust and guarantees correct execution, regardless of the failure of parts of the system. **Napier** programs are compiled into Persistent Abstract Machine (PAM) code and it is an invocation of the PAM that executes the programs. **One of the most notable features of the PAM [2,3] is that it is I Bn Implementing Persistent Object Bases: Principles and Practice Proc.** Fourth International Workshop on Persistent Object Systems, Martha s Vineyard, Massachusetts, September 1990 , A. **Dearle**, G. M. Shaw and S. **B**. Zdonik (Eds. **pp.**103 113 (Morgan Kaufmann Publishers, Inc. **San** ....

Connor R., Brown A.L., Carrick R., Dearie A. & Morrison R. "*The Persistent Abstract Machine*". In "Persistent Object Systems". Workshops in Computing, Rosenberg J. & Koch D. (eds.) pp. 279-288 (Springer-Verlag, 1990.

Cache Coherency and Storage Management in a Persistent Object.. - Koch, al. (1990) (13 citations) (Correct)

....in an environment that is robust and guarantees correct execution, regardless of the failure of parts of the system. **Napier** programs are compiled into Persistent Abstract Machine (PAM) code and it is an invocation of the PAM that executes the programs. **One of the most notable features of the PAM [2,3] is that it is I Bn Implementing Persistent Object Bases: Principles and Practice Proc.** Fourth International Workshop on Persistent Object Systems, Martha s Vineyard, Massachusetts, September 1990 , A. **Dearle**, G. M. Shaw and S. **B**. Zdonik (Eds. **pp.**103 113 (Morgan Kaufmann Publishers, Inc. **San** ....

Morrison R., Brown A.L., Carrick R., Connor R. & Dearle A. "*The Persistent Abstract Machine*". Universities of Glasgow & St Andrews Persistent Programming Research Report 59-88 (1988).

Addressing Large Distributed Collections of Persistent Objects: .. - Eliot Moss (1989) (9 citations) (Correct)

....and easy to justify. **Systems that have taken the large virtual memory approach include the Intel 432 [Organick, 1983; Intel Corporation, 1981] the IBM RT [Chang and Mergen, 1988] the Bubba project [Copeland et al. 1988] CPOMS and related systems [Brown and Cockshott, 1986; Cockshott, 1987; Connor et al. 1989] and MONADS [Keedy and Rosenberg, 1989] Chang and Mergen, 1988] and [Copeland et al. 1988] deal more especially and completely with problems of concurrent access and reliability (transaction support) In particular, past objections to flat stores that have been addressed with some success ....**

R. Connor, A. Brown, R. Carrick, A. Dearle, and R. Morrison. *The persistent abstract machine*. In Persistent Object Systems: Their Design, Implementation, and Use (University of Newcastle, NSW, Australia, Jan. 1989), Department of Computer Science, pp. 80--95.

Working with Persistent Objects: To Swizzle or Not to Swizzle - Moss (1992) (79 citations) (Correct)

....objects from the store server format to a faster in memory format. **POSs** and storage managers include the Exodus storage manager [23, 32] O 2 [29] and Mneme 4 [33, 34, 35] Mneme is the POS used for this study. **There have also been a number of designs related to virtual memory such as [2, 3, 36, 37, 38].** Object servers include ObServer [39] and Gemstone [40] We know of no prior studies of swizzling performance, and hence can offer no comparison with directly related work. **Published** OODB benchmarks and performance studies include [41, 42, 43, 44, 45, 46] 1.2 Simplifying assumptions To study ....

R. Connor, A. Brown, R. Carrick, A. Dearle, and R. Morrison, "*The persistent abstract machine*," in Rosenberg and Koch [57], pp. 353--366.

Delivering the Benefits of Persistence to System Construction and.. - Cutts (1992) (15 citations) (Correct)

....Using Persistence to Optimise Type Checking Type systems are well understood as mechanisms which impose static safety constraints upon a program. **Within** a persistent programming environment the type system provides all the data modelling and protection facilities for the environment. **Elsewhere [CBC 90] it has been demonstrated that not all constraints on data may be captured statically.** This leads to a judicious mixture of type checking times being employed to suit the particular

requirements of an application. **The** spread of times allows a balance between static safety in large applications ....

....of type representations may also be cached in the environment and shared across programs. **The** only type system operation traditionally available outside the compilation environment is that of type equivalence. **In order to maintain execution speed the efficiency of this operation is optimised [CBC 90] The availability of a single set of 10 type system operations and the sharing of type representations between programs simplify the use of any type system operation in any of the construction, compilation, linking or execution contexts.** The implementation of complex type operations such as ....

[Article contains additional citation context not shown here]

R.C.H. Connor, A.L. Brown, R. Carrick, A. Dearle and R. Morrison "*The Persistent Abstract Machine*" In Persistent Object Systems, Springer-Verlag (1990) pp 353-366.

Delivering the Benefits of Persistence to System Construction and.. - Cutts (1992) (15 citations) (Correct)

....produces a low level intermediate code representation. **A** second pass is required to convert this representation into target machine code. **This** pass is performed incrementally at the end of the compilation of each procedure. **There are two existing implementations of the second pass for PAM [BCC 88] code and for SPARC [Sun87] code.** 4.3.2 Building the Compiler on the Construction Architecture When no compiler is available in the persistent environment, one must be constructed using an external program construction environment. **The** binding styles described in Section 4.2 are therefore not ....

A.L. Brown, R. Carrick, R.C.H. Connor, A. Dearle and R. Morrison "*The Persistent Abstract Machine*" Universities of Glasgow and St Andrews Report PPRR-59-88 (1988).

A Model for Persistent Shared Memory Addressing in.. - Amaral, Jacquemot, Lea (1992) (4 citations) (Correct)

....An example of such a system is an implementation of Napier [26] developed by Vaughan et al. 32] over the Mach [1] microkernel. **Napier** is a persistent type system developed by the PISA project. **It consists of a language and a persistent store and executes on a persistent abstract machine (PAM) [12].** Objects in this implementation of Napier are assigned permanent virtual addresses and persistent applications execute against a single global address space. 3 The persistent context space model The persistent context space model is intended to be used by systems that need to manipulate address ....

R. Connor, R. Carrick, A. Dearle, and R. Morrison. *The persistent abstract machine*. In Proceedings of the 3rd Workshop in Persistent Object Systems, Newcastle, Australia, 1989.

Combining Mobile Agents with Persistent Systems.. - Silva, Atkinson (Correct)

....normal values in the language, they offer the same properties as any other value (in particular type safety) Heterogeneity Agents built on one machine [must] execute on other machines . **Napier88 is compiled into an intermediate language (byte code) and then interpreted by an abstract machine [18].** Napier88 currently runs on SunSparc with SunOS 4.1, Alphas with OSF 1 and Pentium with Solaris. **The** Napier88 abstract machine is sufficiently high level so that it can be easily ported to new machine architectures and operating systems. **The** Macintosh and Linux are currently being considered. ....

R.C.H. Connor, A.L. Brown, R. Carrick, A. Dearle, and R. Morrison. *The persistent abstract machine*. In J. Rosenberg and D.M. Koch, editors, Persistent Object Systems, pages 353--366. Springer-Verlag, 1990.

Linguistic Support for Persistent Modules and Capabilities - Rosenberg, Hitchens (Correct)

....we hope to be able to make use of the existing tools developed for Napier88, including parts of the compiler, and this is simplified if MPL has a similar style to Napier88. **Finally we intend to build a first implementation of 3 MPL above the Napier88 store and persistent abstract machine [5] which is designed to support a Napier88 style of language.** The key differences between Napier88 and MPL relate to the latter s support for modules and capabilities. **In** Napier88 the basic structuring tool is the procedure; procedures are first class data types and are used to generate persistent ....

Brown, A. L., Connor, R. C. H., Carrick, R., Dearle, A. and Morrison, R. "*The Persistent Abstract Machine*", Universities of Glasgow and St. Andrews, Persistent Programming Research Report PPRR-59-88, 1988.

Using the Flask Architecture to Build Distributed Applications - Graham Kirby Richard (1995) **Self-citation (Morrison)** (Correct)

No context found.

* Connor, R.C.H., Brown, A.L., Carrick, R., Dearle, A. & Morrison, R. "*The Persistent Abstract Machine*". In Persistent Object Systems, Rosenberg, J. & Koch, D.M. (ed). Springer-Verlag, Proc. 3rd International Workshop on Persistent Object Systems,

Newcastle, Australia (1990) pp 353-366.

PMOS Revisited - Fred Brown And **Self-citation (Brown)** (Correct)

No context found.

Connor, R.C.H., Brown, A.L., Carrick, R., Dearle, A. & Morrison, R. "*The Persistent Abstract Machine*". 3rd International Workshop on Persistent Object Systems, Newcastle, N.S.W., (January 1989), 80-95. In Persistent Object Systems (Eds. J.Rosenberg & D.Koch). SpringerVerlag, 353-366.

A Layered Persistent Architecture for Napier88 - Brown, Dearle, Morrison.. **Self-citation (Brown Dearle Morrison)** (Correct)

....integer key to identify the size of stack elements to manipulate and the rules for performing equality. **The** integer keys are made available to polymorphic code as part of the static environment provided by the block retention. **The Napier88 compilation system maps programs onto an abstract machine[bro88a].** The abstract machine is based on block retention and is responsible for implementing those primitives necessary to support the polymorphism and abstract data types. **In** turn, the abstract machine views the persistent object store as a single heap of persistent objects that is assumed to be a ....

Brown A.L., Carrick R., Connor R.C.H., Dearle A. & Morrison R. *The Persistent Abstract Machine*. Universities of Glasgow and St.Andrews PPRR-59, Scotland, 1988.

Types and Polymorphism in Persistent Programming Systems - Connor (1990) (16 citations) **Self-citation (Connor)** (Correct)

.... **addressing problem 113 The problems of implementing parametric polymorphism have been successfully addressed by the implementors of the Ada programming environment [Ich83] the Functional Abstract Machine [Car83] designed primarily to implement ML, and the Persistent Abstract Machine [BCC88, CBC89], designed primarily to implement Napier88.** The different requirements of these languages have led to three quite different solutions. **The** Ada solution is textual, the ML FAM solution is uniform, and the Napier88 PAM solution is partly tagged. **Each** solution is examined in turn, along with reasons ....

....model is less efficient than that of textual polymorphism, but continues to work in the presence of first class procedures. **This scheme, without reverse encapsulation, has been successfully implemented in the Napier88 system, with architectural support provided by the Persistent Abstract Machine [BCC88, CBC89].** 5.2.5 Conclusions Three different models of implementing parametric polymorphism have been discussed in depth. **Of** particular interest is the suitability of the models for use in a persistent system. **Textual** polymorphism depends upon the compiler producing different versions of code for ....

[Article contains additional citation context not shown here]

A.L. Brown, R. Carrick, R.C.H Connor, A. Dearle and R. Morrison "*The Persistent Abstract Machine*" University of St Andrews PPRR-59-88 (1988)

Persistent Object Stores - Brown (1988) (24 citations) **Self-citation (Brown)** (Correct)

....is described that permits the complete layered architecture to be efficiently implemented with or without the support of special purpose hardware. **To demonstrate the feasibility of the architecture, Chapter 6 presents a description of the first implementation used by the programming language Napier[bro88a,mor88].** This is supplemented in Chapter 7 by a description of how the architecture could be used to support experiments with distribution. **To** conclude the description of the layered architecture, Chapter 8 presents a technique for tuning the architecture layers to reflect a particular choice of system ....

....layers implementing the store. **The** abstract machine provides an instruction set suitable for executing PAIL programs. **It** also provides any primitives required by PAIL to support operations such as concurrency, transactions and distribution. **A description of the abstract machine is given elsewhere[bro88a].** The abstract machine is able to access the architecture s store layers. **However**, it only supports abstract instructions that manipulate persistent objects. **Thus**, any details concerning the storage of persistent objects are not visible to architecture layers, such as PAIL, that may use the ....

[Article contains additional citation context not shown here]

Brown A.L., Carrick R., Connor R.C.H., Dearle A. & Morrison R. *The Persistent Abstract Machine*. Universities of Glasgow and St.Andrews PPRR-59, Scotland, 1988.

Incremental Garbage Collection of a Persistent Object.. - Munro, Brown.. (1998) (1 citation) **Self-citation (Brown Morrison)** (Correct)

No context found.

Connor, R.C.H., Brown, A.L., Carrick, R., Dearle, A. & Morrison, R. "*The Persistent Abstract Machine*". 3rd International Workshop on Persistent Object Systems, Newcastle, N.S.W., (January 1989), 80-95. In Persistent Object Systems (Eds. J.Rosenberg & D.Koch). Springer-Verlag, 353-366.

Operating System support for Java - Dearle, Hulse, Farkas (1996) (3 citations) **Self-citation (Dearle)** (Correct)

....system it was found that the use of copy out techniques borrowed from generational garbage collection considerably reduced the work of garbage collecting the global heap. **It is expected that similar results will be found in the case of Java whose architecture is very similar to that of Napier88 [8]. 4. Binding** In all persistent systems, some mechanism must be provided to permit objects to bind to other objects in the persistent store. **In** systems such as Napier88 this is provided by the environment mechanism [9] In Grasshopper, any locus can generate an arbitrary address in the container ....

Connor, R., Brown, A., Carrick, R., Dearle, A. and Morrison, R. "*The Persistent Abstract Machine*", Proceedings of the Third International Workshop on Persistent Object Systems, Newcastle, Australia, Springer-Verlag, pp. 353-366, 1989.

Using C as a Compiler Target Language for Native Code.. - Bushell Dearle **Self-citation (Brown Dearle)** (Correct)

.... **of the following options: whether the store is directly mapped with page granularity, or swizzed with object granularity; whether the code in the store is native code or interpreted PAM code; whether the code in the store uses the Octopus model [8] or the original PAM frame model [4] ; and .** whether the system will run on the Sun SPARC architecture or the DEC Alpha AXP architecture. **We** plan to measure the performance of the OO7 benchmark [3] under all combinations of the above options, comparing object swizzled and page mapped stores. **Acknowledgements** We would like thank the ....

Connor, R., Brown, A., Carrick, R., Dearle, A. and Morrison, R. "*The Persistent Abstract Machine*", Proceedings of the Third International Workshop on Persistent Object Systems, Newcastle, Australia, Springer-Verlag, pp. 353-366, 1989.

Using C as a Compiler Target Language for Native Code.. - Bushell Dearle **Self-citation (Brown Dearle)** (Correct)

....Algol like language with nested scope. **Although** C is block structured, it does not support nested functions and therefore some mechanism must be provided in the generated code to support scope. **The interpreted version of Napier88 executes on a machine known as the Persistent Abstract Machine (PAM) [2] .** In this implementation, procedure values or closures are implemented as a pair of pointers: one to an object containing code, the code vector, the other to the activation record of the defining procedure, the static link. **Since** procedures are first class values, they may escape the scope in ....

.... **= x ; save restart point in stack frame fixed register local frame[ResumeAddress] restart start ; return garbage collect request return unwind and continue ; restart: restore x x = fixed register local frame[x offset] goto create ; repeat attempt to create S S[2] = x ; initialise S .** update local frame to point at caller fixed register local frame = fixed register local frame[DLink] return OK ; normal completion Figure 4: The C function implementing the Napier88 procedure shown in Figure 5. **When** they start executing, ...

Brown, A. L., Carrick, R., Connor, R. C. H., Dearle, A. and Morrison, R. "*The Persistent Abstract Machine*", Universities of Glasgow and St Andrews, Technical Report PPRR-59-88, 1988.

Persistent Operating System Support for Java - Dearle, Hulse, Farkas (1996) (5 citations) **Self-citation (Dearle)** (Correct)

....system it was found that the use of copy out techniques borrowed from generational garbage collection considerably reduced the work of garbage collecting the global heap. **It is expected that similar results will be found in the case of Java whose architecture is very similar to that of Napier88 [8].** However, at the time of writing, this technique has not been implemented in the prototype and therefore further results are not currently available. 4. **Binding** In all persistent systems, some mechanism must be provided to permit objects to make bindings to other objects in the persistent ....

Connor, R., Brown, A., Carrick, R., Dearle, A. and Morrison, R. "*The Persistent Abstract Machine*", Proceedings of the Third International Workshop on Persistent Object Systems, Newcastle, Australia, Springer-Verlag, pp. 353-366, 1989.

An Ad-Hoc Approach to the Implementation of Polymorphism - R. Morrison, A. Dearle.. (1991) (31 citations) (Correct)

No context found.

Brown, A.L., Carrick, R., Connor, R.C.H., Dearle, A. & Morrison, R. "*The Persistent Abstract Machine*". PPRR-59-88. Universities of St Andrews and Glasgow (1988).

On the Integration of Concurrency, Distribution and Persistence - Munro (1993) (4 citations) (Correct)

No context found.

Connor, R.C.H., Brown, A.L., Carrick, R., Dearle, A. & Morrison, R. "*The Persistent Abstract Machine*". 3rd International Workshop on Persistent Object Systems, Newcastle, N.S.W., (January 1989), 80-95. In Persistent Object Systems (Eds. J.Rosenberg & D.Koch). Springer-Verlag, 353-366.

On the Integration of Concurrency, Distribution and Persistence - Munro (1993) (4 citations) (Correct)

No context found.

Brown A.L., Carrick R., Connor R.C.H., Dearle A. & Morrison R. *The Persistent Abstract Machine*. Universities of Glasgow and St.Andrews PPRR-59, Scotland, 1988.

Java as Persistent Glue - Francis Vaughan (Correct)

No context found.

Brown, A., Carrick, R., Conner, R., Dearle, A., Morrison, R. (1988). *The Persistent Abstract Machine*. Universities of Glasgow and St Andrews. PPRR-59-88.

Modelling Recovery in Database Systems - Scheuerl (1997) (Correct)

No context found.

Connor, R.C.H., Brown, A.L., Carrick, R., Dearle, A. & Morrison, R. "*The Persistent Abstract Machine*". 3rd International Workshop on Persistent Object Systems, Newcastle, N.S.W., (January 1989) pp 80-95. In Persistent Object Systems (Eds. J. Rosenberg & D. Koch). Springer-Verlag pp 353-366. 137

Reflection and Hyper-Programming in Persistent Programming Systems - Kirby (1992) (21 citations) (Correct)

No context found.

Connor, R.C.H., Brown, A.L., Carrick, R., Dearle, A. & Morrison, R. "*The Persistent Abstract Machine*". In Persistent Object Systems, Rosenberg, J. & Koch, D.M. (ed), Springer-Verlag (1990) pp 353-366.

First 50 documents

15 citations found. Retrieving documents...

R. L. Cooper, M. P. Atkinson, A. Dearle and D. Abderrahmane, *Constructing database systems in a persistent environment*, Proc. 13th VLDB Conference, 1987, pp. 117–125.

This paper is cited in the following contexts:

A Modular Compiler Architecture for a Data Manipulation Language - Embury, Gray (1997) (Correct)

.... **support incremental development of code, are examples of such systems [13] as are database management systems (DBMSs) A callable compiler allows the implementation of embedded query languages, browsers and schema evolution capabilities, without sacri cing the security of static type checking [6].** However, as DBMS technology has advanced, the increasing need to manipulate not only the structure of data at run time, but also its semantics, has meant that the ability to invoke compiler functions on the y has become important for DBMSs themselves, and not just for the languages which ....

R.L. Cooper, M.P. Atkinson, A. Dearle, and D. Abderrahmane. *Constructing Database Systems in a Persistent Environment*. In P.M. Stocker, W. Kent, and P. Hammersley, editors, Proceedings of the 13th International Conference on Very Large Databases, pages 117-125, Brighton, 1987. Morgan Kaufmann Publishers, Inc.

Type-Safe Linguistic Run-time Reflection A Practical Perspective - Cooper, Kirby (1994) (3 citations) (Correct)

....has been put is for program modules which are required to adapt themselves to varying types of input. **This mechanism has been used extensively in Napier88 and its predecessor language PS algol [18] Among the uses have been the PS algol browser [19] and a strongly typed relational system [20].** The former allows the user to navigate about values in the PS algol persistent store, displaying an automatically generated menu for the current value, from which the user can traverse to a related value. **The** browser works from a template source module for the menu which it fills in with type ....

Cooper RL, Atkinson MP, Dearle A, Abderrahmane D. *Constructing Database Systems in a Persistent Environment*. In: Proc. 13th International Conference on Very Large Data Bases, 1987, pp 117-125

The Napier88 Persistent Programming Language and.. - Morrison, Connor.. (1999) (Correct)

....as a procedure by any program. **Type** safe linguistic reflection has been used to attain high levels of genericity [87, 88] and accommodate changes in systems [84, 89] two examples of these are given below. **It has also been used to implement data models [62, 63, 90] optimise implementations [91 93] and validate specifications [94, 95] The importance of the technique is that it provides a uniform mechanism for software production and evolution.** A formal description of linguistic reflection is given in [83] The example in Figure 30 shows a simple generator which produces code to write out ....

Cooper RL, Atkinson MP, Dearle A, Abderrahmane D. *Constructing Database Systems in a Persistent Environment*. In: Proc. 13th International Conference on Very Large Data Bases, 1987, pp 117-125

A Modular Compiler Architecture for a Data Manipulation Language - Embury, Gray (Correct)

.... **support incremental development of code, are examples of such systems [11] as are database management systems (DBMSs) A callable compiler allows the implementation of embedded query languages, browsers and schema evolution capabilities, without sacrificing the security of static type checking [4].** However, as DBMS technology has advanced, the increasing need to manipulate not only the structure of data at run time, but also its semantics, has meant that the ability to invoke compiler functions on the fly has become important for DBMSs themselves, and not just for the languages which ....

R.L. Cooper, M.P. Atkinson, A. Dearle, and D. Abderrahmane. *Constructing Database Systems in a Persistent Environment*. In P.M. Stocker, W. Kent, and P. Hammersley, editors, Proceedings of the 13th VLDB Conference, pages 117–125, Brighton, 1987. Morgan Kaufmann Publishers, Inc.

The ADAMS Database Language - Pfaltz, French, Grimshaw, Son.. (1989) (1 citation) (Correct)

....in traditional programming languages. **In** many languages, such as Algol and Pascal [JeW75] they are a kind of stepchild which is explicitly disavowed by the parent language In others, only inherently sequential stream I O is supported. **None, with the possible exception of persistent Pascal [BuA86, CAD87], employ a computational model in which the process is coequal with a permanent database from which specific data items are directly accessible.** ADAMS was created in response to these kinds of perceived deficiencies. **This** report represents the combined design efforts of its authors over a three ....

R. L. Cooper, M. P. Atkinson, A. Dearle and D. Abderrahmane, *Constructing Database Systems in a Persistent Environment*, Proc. 13th VLDB Conf., Brighton, England, Sep. 1987, 117-125.

Implementation and Evaluation of Scalability Techniques in the ECO .. - Haahr (1998) (Correct)

....Compilation and Runtime Type Information A way of solving the problem with runtime generated notify constraints could be to use a runtime compiler instead of a preprocessor. **Runtime** compilation lets a program, such as the minefield code shown above, compile and link code at runtime. **PS Algol [CADA87] is a language with support for runtime compilation in form of a library function that takes a source code (in form of a string) and returns a procedure.** In the context of notify constraints, such a function could be used to map notify constraint strings generated at runtime to executable code. **A** ....

R. L. Cooper, M. P. Atkinson, A. Dearle, and D. Abderrahmane. *Constructing database systems in a persistent environment.* In Peter M. Stocker and William Kent, editors, Proceedings of the Thirteenth International Conference on Very Large Databases, pages 117--126, Brighton, 9 1987.

Delivering the Benefits of Persistence to System Construction and.. - Cutts (1992) (15 citations) (Correct)

....a type safe mechanism for the production and evolution of programs and data in a persistent environment. **In current systems it has been used to attain high levels of genericity [SFS 90] accommodate changes in systems [DB88,DCK89] implement data models [Coo90a,Coo90b] optimise implementations [CAD 87,FS91] and validate specifications [FSS92,SSF92] The focus of interest here is the manner in which linguistic reflection may be used to support the programming process entirely within the persistent environment.** To achieve this, a particular style of linguistic reflection known as run time ....

....compiler in the construction and execution of new programs that manipulate the persistent environment. **This is a form of reflection [Mae87] and has been used to attain high levels of genericity [SFS 90] accommodate changes in systems [DB88] implement data models[Coo90a] optimise implementations[CAD 87] and validate specifications[FSS92] A number of persistent languages and their associated object stores have already been implemented.** These include PS algol [PS88] Napier88 [MBC 89] Galileo [ACO85] DBPL[MS89] Staple [DM90] and Quest [Car89] The compilers for these languages construct ....

R.L. Cooper, M.P. Atkinson, A. Dearle and D. Abderrahmane *"Constructing Database Systems in a Persistent Environment"* In Proc. 13th International Conference on Very Large Data Bases, (1987) pp 117-125.

Basic Database Concepts in ADAMS (Advanced DAta Manipulations.. - John Pfaltz (Correct)

....not, been assigned for x. **The** ADAMS expression assign (x.a, string) assigns string as the value of x. **a provided, of course, that string is actually in the co domain (i.e. satisfies the defining LEX expression) These ways of referencing co domain values correspond to getDomVal and putDomVal in [CAD87].** We observe that many computational processes using ADAMS as their database interface do not manipulate strings. **They** typically manipulate numeric values, reals and integers. **A** more characteristic processor ADAMS interface would look like: X : decode(x.a) assign (x.a, encode(Y) where X and ....

R. L. Cooper, M. P. Atkinson, A. Dearle and D. Abderrahmane, *Constructing Database Systems in a Persistent Environment*, Proc. 13th VLDB Conf., Brighton, England, Sep. 1987, 117-125.

A Review of the Rationale and Architectures of PJama: a.. - Atkinson, Jordan (2000) **Self-citation (Atkinson)** (Correct)

....5 languages do not have the investment that leads to rich libraries and high quality implementations. **The** first example, PS algol [13] pioneered orthogonal persistence, and was used extensively in a few contexts. **For example, it was used for data model [99, 132] and query language experiments [52], and at ICL to implement a set of process modelling tools.** 3 The latter should have proved a useful test of the hypothesis, but the experiences were not analysed and published, and are no longer recent. **Napier88** [159] involved an experimental type system, the unfamiliarity of which limited its ....

....in full. **They** cover very nearly all of the requirements on a good development programming language and on a good database management system (DBMS) Three omission deserve comment. **9 Query Systems Our past experience suggests that it is reasonable to expect to build these within the language [99, 132, 52, 21, 24, 124, 162].** We agree that bulk operations and queries are useful. **But** they are useful over any large, regularly structured data structure, irrespective of its location and longevity. **Indexes** Again, we hold that these are best built within the language. **They** then gain the portability of implementations ....

R.L. Cooper, M.P. Atkinson, A. Dearle, and D. Abderrahmane. *Constructing Database Systems in a Persistent Environment.* In Proceedings of the 13th International Conference on Very Large Data Bases, pages 117--125, 1987.

On the Construction of Persistent Programming Environments - Dearle (1988) (12 citations) **Self-citation (Dearle)** (Correct)

....Such compilers have proved to be of great utility in providing a richer programming environment; whilst maintaining a strict type regime. **Callable compilers have been used to provide an object browser [dea88b] discussed in the next chapter and an adaptive relational database implementation [coop87].** 6.6 Interactive Compilers A Callable Compiler which reads from the users console and immediately executes the code produced is known as a compile and go compiler. **Such** a compiler may be used in a way which resembles a shell [bou78,joy80] This shell is of much greater power due to the support ....

....The production of a virtual image using this technique is much cheaper than hand building a virtual image or even writing a program to produce a custom made one. 7. **11 Adaptive Databases The technology used in the browser has also been used in the production of a relational database system [coop87].** Traditionally, databases are implemented by creating a canonical relation structure [cod70] Relations introduced by the programmer of the system are then mapped onto this canonical representation. **Relations** then require a level of secondary interpretation at run time. **The** database system ....

Cooper R.L.,Atkinson M.P., Dearle A. & Abderrahamane A. *Constructing Database Systems in a Persistent Environment*. Proc VLDB 1987, Brighton England, (1987).

Experiences in Database System Implementation Using a Persistent.. - Hanson (Correct)

No context found.

R. L. Cooper, M. P. Atkinson, A. Dearle and D. Abderrahmane, '*Constructing database systems in a persistent environment*', Proc. 13th VLDB Conference, 1987, pp. 117--125.

Linguistic Reflection in Java - Kirby, Morrison, Stemple (1998) (3 citations) (Correct)

No context found.

R. L. Cooper, M. P. Atkinson, A. Dearle and D. Abderrahmane, '*Constructing Database Systems in a Persistent Environment*', Proceedings of 13th International Conference on Very Large Data Bases, 117125 (1987).

Mechanisms for Controlling Evolutions in Persistent.. - Morrison, Connor.. (1993) (2 citations) (Correct)

No context found.

Cooper, R.L., Atkinson, M.P., Dearle, A. & Abderrahmane, D. "*Constructing Database Systems in a Persistent Environment*". 13th VLDB, Brighton, UK (1987), pp 117-126.

Integrating Reflection, Strong Typing and Static Checking - Stemple, Morrison, Kirby.. (1993) (1 citation) (Correct)

No context found.

Cooper, R.L., Atkinson, M.P., Dearle, A. & Abderrahmane, D. "*Constructing Database Systems in a Persistent Environment*". In Proc. 13th International Conference on Very Large Data Bases (1987) pp 117-125.

Reflection and Hyper-Programming in Persistent Programming Systems - Kirby (1992) (21 citations) (Correct)

No context found.

Cooper, R.L., Atkinson, M.P., Dearle, A. & Abderrahmane, D. "*Constructing Database Systems in a Persistent Environment*". In Proc. 13th International Conference on Very Large Data Bases (1987) pp 117-125.

41 citations found. Retrieving documents...

M. P. Atkinson, M. J. Jordan, L. Daynes, and S. Spence. *Design issues for persistent Java: A type-safe*, object-oriented, orthogonally persistent system. In Persistent Object Systems (POS), May 1996.

This paper is cited in the following contexts:

First 50 documents

Lazy Modular Upgrades in Persistent Object Stores - Boyapati, Liskov, Shrira.. (2003) (Correct)

....modi cations of class de nitions are applied; all object instances are converted (eagerly or lazily, but once and forever) to conform to the latest schema. **The schema evolution approach is used in Orion [7] OTGEN [40] O2 [29, 52] GemStone [17, 48] Objectivity DB [47] Versant [50] and PJama [6, 5] systems, and is the only approach available in commercial RDBMS.** An extensive survey of the previous schema evolution systems can be found in [30] None of the previous schema evolution systems provide a way of executing upgrades eciently both in space and time, while allowing programmers to ....

M. P. Atkinson, M. J. Jordan, L. Daynes, and S. Spence. *Design issues for persistent Java: A type-safe*, object-oriented, orthogonally persistent system. In Persistent Object Systems (POS), May 1996.

Ownership Types for Object Encapsulation - Boyapati, Liskov (2003) (1 citation) (Correct)

....This section shows how ownership types and e ects clauses can be used to enable modular reasoning about the correctness of upgrades in a persistent object store. **The** desire to achieve such reasoning was the motivation for our work on ownership types for encapsulation. **A persistent object store [46, 5, 9, 17, 18, 29, 56] contains conventional objects similar to what one might nd in an object oriented language such as Java.** Applications access persistent objects within atomic transactions, since this is necessary to ensure consistency for the stored objects; transactions allow for concurrent access and they mask ....

M. P. Atkinson, M. J. Jordan, L. Daynes, and S. Spence. *Design issues for persistent Java: A type-safe*, object-oriented, orthogonally persistent system. In Persistent Object Systems (POS), May 1996.

Ownership Types and Safe Lazy Upgrades in Object-Oriented.. - Boyapati, Liskov, Shrira (2002) (Correct)

....later and shared with other applications. **The database acts as an extension of an object oriented programming language such as Java, allowing programs access to long lived objects in a manner analogous to how they manipulate ordinary objects whose lifetime is determined by that of the program** [37, 24, 13, 43, 12, 5]. The objects stored in an OODB may live a long time and as a result there may be a need to upgrade them, that is, change their code and storage representation. **An** upgrade can improve an object s implementation, to make it run faster or to correct an error; extend the object s interface, e.g. by ....

....object via multiple potentially incompatible interfaces and are very di erent from the eciency and correctness issues in the evolutionbased upgrade systems. **The schema evolution approach is used in Orion [6] OTGEN [36] O2 [24, 53] GemStone [12, 45] Objectivity DB [44] Versant [49] and PJama [5, 4] systems, and is the only approach available in any commercial RDBMS.** None of the existing schema evolution systems provides both expressive and ecient upgrades. **Furthermore**, none bases the correctness of the upgrade system on the property of encapsulation or ownership types. **Work** in O2 uses ....

M. P. Atkinson, M. J. Jordan, L. Daynes, and S. Spence. *Design issues for persistent Java: A type-safe*, object-oriented, orthogonally persistent system. In Persistent Object Systems (POS), May 1996.

Ownership Types and Safe Lazy Upgrades in Object-Oriented.. - Boyapati, Liskov, Shrira (2002) (Correct)

....later and shared with other applications. **The database acts as an extension of an object oriented programming language such as Java, allowing programs access to long lived objects in a manner analogous to how they manipulate ordinary objects whose lifetime is determined by that of the program** [37, 24, 13, 43, 12, 5]. The objects stored in an OODB may live a long time and as a result there may be a need to upgrade them, that is, change their code and storage representation. **An** upgrade can improve an object s implementation, to make it run faster or to correct an error; extend the object s interface, e.g. by ....

....object via multiple potentially incompatible interfaces and are very di#erent from the e#ciency and correctness issues in the evolutionbased upgrade systems. **The schema evolution approach is used in Orion [6] OTGEN [36] O2 [24, 53] GemStone [12, 45] Objectivity DB [44] Versant [49] and PJama [5, 4] systems, and is the only approach available in any commercial RDBMS.** None of the existing schema evolution systems provides both expressive and e#cient upgrades. **Furthermore**, none

bases the correctness of the upgrade system on the property of encapsulation or ownership types. **Work** in O2 uses ....

M. P. Atkinson, M. J. Jordan, L. Daynes, and S. Spence. *Design issues for persistent Java: A type-safe*, object-oriented, orthogonally persistent system. In Persistent Object Systems (POS), May 1996.

Towards An Extensible Virtual Machine - Boyapati (2002) (Correct)

....All objects reachable from persistent roots are automatically stored in persistent storage by the system the rest of the objects are garbage collected. **Since** Java has no persistence model built into it, many research systems have been proposed that add persistence to Java. **These include PJama [3], JPS [6] GemStone J [17] and PSEJ [42] Of these, only PSEJ is JVM compatible.** PSEJ uses the technique of bytecode rewriting. **Bytecode** rewriting has become an established technique for extending Java. **But** PSEJ has severe problems both in terms of its semantics and its performance. **For** example, ...

....system must provide a mechanism for upgrading the persistent objects. **These** upgrades involve 2 changes to the code implementing the persistent objects, as well as changes to the persistent objects themselves. **Much** research has been done on software evolution in persistent object systems. **PJama [3, 12], JPS [6, 31] and GemStone [17, 40] support some form of software evolution using modified JVMs.** Software evolution cannot be implemented on unmodified JVMs. 2.4 Reflective Interface for Java Metaobject protocols [23] o#er a principled way of extending the behavior of programs. **Metaobjects** can ....

M. P. Atkinson, M. J. Jordan, L. Daynes, and S. Spence. *Design issues for persistent Java: A type-safe*, object-oriented, orthogonally persistent system. In Persistent Object Systems (POS), May 1996.

Towards An Extensible Virtual Machine - Boyapati (2002) (Correct)

....All objects reachable from persistent roots are automatically stored in persistent storage by the system the rest of the objects are garbage collected. **Since** Java has no persistence model built into it, many research systems have been proposed that add persistence to Java. **These include PJama [3], JPS [6] GemStone J [17] and PSEJ [42] Of these, only PSEJ is JVM compatible.** PSEJ uses the technique of bytecode rewriting. **Bytecode** rewriting has become an established technique for extending Java. **But** PSEJ has severe problems both in terms of its semantics and its performance. **For** example, ...

....system must provide a mechanism for upgrading the persistent objects. **These** upgrades involve 2 changes to the code implementing the persistent objects, as well as changes to the persistent objects themselves. **Much** research has been done on software evolution in persistent object systems. **PJama [3, 12], JPS [6, 31] and GemStone [17, 40] support some form of software evolution using modi ed JVMs.** Software evolution cannot be implemented on unmodi ed JVMs. 2.4 Re ective Interface for Java Metaobject protocols [23] o er a principled way of extending the behavior of programs. **Metaobjects** can be ....

M. P. Atkinson, M. J. Jordan, L. Daynes, and S. Spence. *Design issues for persistent Java: A type-safe*, object-oriented, orthogonally persistent system. In Persistent Object Systems (POS), May 1996.

JPS: A Distributed Persistent Java System - Boyapati (1998) (1 citation) (Correct)

....and implementation of a Java Persistent Store called JPS. **JPS** is an ecient distributed persistent Java system built on top of the Thor [LAC 96, CALM97] object oriented database. **There** are many other research and commercial e orts underway to add persistence to Java. **These include PJama [AJDS96] which is being built by Sun and the University of Glasgow, the Persistent Storage Engine for Java [PSE] developed by ObjectStore, and GemStone J [Gem] developed by GemStone.** Also, Oracle is planning to put Java into their next version of the Oracle8 database server, release 8.1. **However**, our ....

....have to depend on the paging mechanism of underlying operating system to manage the cache. **One** possible approach to integrate Java into Thor was to appropriately modify a standard Java interpreter and incorporate it into the client side of Thor. **This is similar to the approach taken by the PJama [AJDS96] project.** However, since interpreted code runs one to two orders of magnitude slower than compiled code [PTB 97, MMBC97] we abandoned this path. **Another** option was to build a Just In Time (JIT) compiler [JIT] into Thor s client side. **Though** this might have been the ideal solution in the ....

[Article contains additional citation context not shown here]

M. P. Atkinson, M. J. Jordan, L. Daynes, and S. Spence. *Design issues for Persistent Java: A type-safe*, object-oriented, orthogonally persistent system. In Seventh International Workshop on Persistent Object Systems (POS7), Cape May, New Jersey, USA, 1996. Available at http://www.sunlabs.com/research/forest/COM.Sun.Labs.Forest.doc- .external www.papers.html.

Fruitlets - a Kind of Mobile Component - Gertsch (1997) (Correct)

....really movable objects does not carry too much weight. **Java** is very popular at the moment and thus there are a lot of running projects related to the Internet, World Wide Web and mobile agents. **Projects** like PJava (Glasgow Persistent CHAPTER 4.

PROGRAMMING LANGUAGES FOR MOBILE PROGRAMMING 33 Java) [AJDS96] or Active Objects (Doug Lea) 6 will help to improve the language. Java is already strong in the sense of platform neutrality. **Interpreters** for most of the common platforms are available. **Concurrency** constructs are integrated into the language itself, which makes concurrent programming very easy ....

M.P. Atkinson, M.J. Jordan, L. Daynes, and S. Spence. *Design Issues for Persistent Java: a type-safe*, object-oriented, orthogonally persistent system. Technical report, University of Glasgow, February 1996.

The First Experience of Class Evolution Support in PJama - Misha Dmitriev Misha (1998) (5 citations) (Correct)

....In this paper we report on the issues raised when investigating the ways of and developing the mechanisms for class evolution in PJama. **The PJama project, a collaboration between Sun Microsystems Laboratories and the University of Glasgow, is building an orthogonally persistent platform for Java [1, 2].** PJama is the implementation of the programming language Java with mechanisms embedded into the runtime system that support indefinite lifetimes for a program s data. **In** fact, not all the data of some program, but only objects chosen by a PJama programmer persist. **On** the other hand, objects ....

M.P. Atkinson, L. Dayn'es, M.J. Jordan, and S. Spence. *Design Issues for Persistent Java: A Type-Safe*, Object-Oriented, Orthogonally Persistent System. In The Proceedings of the 7th International Workshop on Persistent Object Systems (POS 7), May 1996.

A Selective Protection Scheme for the Java Environment - Hagimont, Krakowiak.. (1996) (2 citations) (Correct)

....callee class (which may or not be co located with the calling class) and (b) what access rights should be attached to the file parameters. 2) Developing really distributed applications with Java implies the integration of persistence in the language. **Work is in progress towards that goal (e.g. Atkinson 96] In our current work, persistence is only achieved through the use of files.** We intend to examine a possible extension of our protection environment to cover persitent Java objects. **Despite** its current limitations, we think that this work is a promising step toward the development of secure ....

M. P. Atkinson, M. J. Jordan, L. Dayns, and S. Spence, *Design Issues for Persistent Java: a type-safe*, object-oriented, orthogonally persistent system, to appear in Proc. of 7th Workshop on Persistent Object Systems (POS-7), 1996.

Transactions for Java - Alex Garthwaite And (1996) (10 citations) (Correct)

....is likely that it was the first such collector. **However** this lack means that at the least Almes techniques will have to be significantly extended to support concurrent collection for Jest. **We are aware of two other efforts to provide generalpurpose persistence for Java: PJama (formerly Pjava) [4, 16, 6] and Gemstone J [8] PJama uses the same model of persistence by reachability that Jest does.** However, their design goals are quite different. **At** the language level, PJama takes the conservative approach of not extending the syntax of Java or its bytecode representation. **Instead**, it provides ....

M. Atkinson, M. Jordan, L. Dayn'es, and S. Spence. *Design Issues for Persistent Java: A Type-Safe*, Object-Oriented, Orthogonally Persistent System. In the Pre-Proceedings of the 7th International Workshop on Persistent Object Systems (POS 7), May 1996.

Persistent Servers + Ephemeral Clients = User Mobility - Alan Dearle University (1997) (2 citations) (Correct)

....is how state is preserved; in general, there are 4 approaches to saving state in Java systems: 1. **manually** writing save and restore code in every application applet, 2. **perform** saving and restoration using (Java) serialisation, 3. **providing persistence at the (Java) virtual machine level [2], and 4.** providing persistence at the address space level. **The** first approach is the traditional solution to persistence: write flattening code by hand for every object class in the system. **Whilst** this is possible for simple data structures it becomes unmanageable in complex applications and has ....

....Internet. **If** this approach is to be followed, techniques such as Farkas OCTOPUS mechanism [4] or the use of weak pointers are also required. **The** next approach to saving state is to provide persistence at the (Java) virtual machine level. **This is the approach followed by Atkinson s PJava group [2].** Whilst we have argued elsewhere that the last approach is better, this approach has many merits in the application domain described in this paper. **It** also addresses many of the shortcomings of the serialisation approach described above. **Providing** persistence at a level lower than the Java ....

Atkinson, M.P., et al. *Design Issues for Persistent Java: a type safe*, object oriented, orthogonally persistent system. in 7th International Conference on Persistent Object Systems. 1996: Springer-Verlag.

A Dynamic, Portable and Safe Approach to Byte-Code.. - Marquez, Zigman, Blackburn (Correct)

....of orthogonally persistent Java (OPJ) is the choice of means by which the Java language semantics are extended to include

persistence. **While Moss and Hosking [19[illegible] outline a number of choices, the approach of modifying the underlying virtual machine has been dominant in the literature until now [Atkinson et al. 1996; Kutlu and Moss 1998; GemStone Systems 1999] In this paper we show that OPJ can be realized without modification to the Java virtual machine or compiler, and furthermore that while this approach is relatively simple, it can outperform an OPJ based on a modified Java virtual machine.** After ....

....case is the use of a modified JVM with advanced introspection support. **There are a number of ways in which standard Java semantics can be transparently extended, including: 2 modifying the virtual machine to directly implement the semantic extension either through the existing byte code set [Atkinson et al. 1996; GemStone Systems 1999] or via additional byte codes [Kutlu and Moss 1998] modifying the virtual machine to implement extended reflection capabilities through which semantic extensions can be implemented [Kutlu and Moss 1998] pre processing source code [Boyland and Catsagna 1997]** ....

[Article contains additional citation context not shown here]

ATKINSON, M. P., JORDAN, M. J., DAYN ES, L., AND SPENCE, S. 1996. *Design issues for Persistent Java: A type-safe*, object-oriented, orthogonally persistent system. In R. CONNOR AND S. NETTLES Eds., Seventh International Workshop on Persistent Object Systems (Cape May, NJ, U.S.A., May 1996), pp. 33–47. Morgan Kaufmann.

Proceedings of the 2nd ECOOP Workshop on.. - Wohlrab.. (1999) (Correct)

....the notion of persistence from the start, such as Grasshopper [DRH 92] although object orientation is not used as the central paradigm of the system. **Some systems add persistence to an existing object system by adding a layer that provides this property, as in the case of the Java platform [AJDS96] This software is in charge of the exchange between the instance area of the system and the secondary storage.** The goal is not to modify the semantics and operation of the existing system. **Main** memory garbage collection of objects is one of the features of the existing system that is retained. ....

M.P. Atkinson, M.J. Jordan, L. Daynes, and S. Spence. *Design issues for persistent Java: a type-safe*, object-oriented, orthogonally persistent system. In Seventh International Workshop on Persistent Object Systems, 1996.

DRASTIC: A Run-Time Architecture for Evolving, Distributed.. - Evans, Dickman (1997) (Correct)

....root, that is known to the persistence mechanism. **Thus**, any object in a DRASTIC process that is reachable, either directly or indirectly, from the persistent root is periodically made persistent automatically. **In our current implementation orthogonal persistence is provided by Persistent Java [AJDS96] Without support for orthogonal persistence, the programmer would be required to explicitly transfer to and from stable storage data structures that should out live the process that manipulates them.** This means that programmers spend a lot of time flattening their complex data structures into ....

....languages provide the software engineer with the ability to constrain the effects of evolving an application in the way DRASTIC s zones and zone contracts do. 8. **6 Persistent Languages Type evolution is more of a problem for programming languages that support persistence [DCBM89, MMM93, AGO88, AJDS96] than for those that do not.** This is because the persistent store contains data which is loaded into type instances at run time. **If** a type is changed, instances of that type may not longer be retrievable from the store unless some form of transformation is performed. **This** can lead to the entire ....

[Article contains additional citation context not shown here]

M. Atkinson, M. Jordan, L. Dayn`es, and S. Spence. *Design issues for persistent Java: a typesafe*, object-oriented, orthogonally persistent system. In Proceedings of The Seventh International Workshop on Persistent Obje ct Systems, Cape May, New Jersey, USA, May 1996.

Supporting Software Evolution in a Distributed, Persistent System - Evans, Dickman (Correct)

....that created them, typically by being written to hard disk. **This is addressed in CORBA via its Object Persistence Service [Obj95d] DRASTIC supports this using the concept of orthogonal persistence by reachability [AM95] and in our current implementation this is provided by Persistent Java [AJDS96] Orthogonal persistence by reachability is modeled by transitive closure from a distinguished object, a persistent root, that is known to the persistence mechanism.** Any object in a DRASTIC process which is reachable, either directly or indirectly, from the persistent root is made persistent ....

M. Atkinson, M. Jordan, L. Dayn`es, and S. Spence. *Design issues for persistent java: a type-safe*, object-oriented, orthogonally persistent system. In Proceedings of The Seventh International Workshop on Persistent Object Systems, Cape May, New Jersey, USA, May 1996.

Transparent Access to Legacy Data in Java - Gruber (Correct)

....be browsed but also queried if one is to provide a query processor and a query engine in the language. **Transparent** access to corporate data can be looked at as an extension to a persistent language. **In this paper, we will assume the existence of a persistence framework such as the one in PJava [2] which provides both persistence by reachibility and a transactional framework.** Our approach is based on the notion of an external object faulter, conceptually very close to the concept of external pagers in operating systems. **The** external object faulter is in charged of fetching or putting away ....

....as an extension to a persistent language. **It may be possible to provide XOFs for non persistent languages, however we feel having a persistent host language simplifies the design and provides potentially more power, especially when one considers transactional persistent languages such as PJava [2].** The key point addressed by external object faulters is that some objects persist outside the standard persistence mechanism of the language. **This** has two consequences. **First**, the language run time cannot be responsible for fetching and saving all object states. **Second**, it is not solely ....

M.P. Atkinson, M.J. Jordan, L. Dayn'es and S. Spence. *Design issues for persistent java: a type-safe*, object-oriented, orthogonally persistent system. In Seventh International Workshop on Persistent Object Systems, 1996.

Evolutionary Data Conversion in the PJama Persistent Language - Dmitriev And Atkinson (1999) (1 citation) **Self-citation (Atkinson)** (Correct)

....system, it may be crucial that its evolution subsystem can guarantee that any reasonable modification of persistent classes (schema) and, consequently, persistent objects, can be performed with minimum effort and without the necessity of rebuilding the whole database from scratch. **PJama [1, 2, 10, 13] is an experimental persistent programming system for the Java programming language.** It has much in common with object oriented database systems used together with Java. **PJama** is being developed as a collaborative project between Glasgow University and Sun Microsystems. **For** PJama, mechanisms are ....

M.P. Atkinson, L. Dayn'es, M.J. Jordan, and S. Spence. *Design Issues for Persistent Java: A TypeSafe*, Object-Oriented, Orthogonally Persistent System. In The Proceedings of the 7th International Workshop on Persistent Object Systems (POS 7), May 1996.

A Review of the Rationale and Architectures of PJama: a.. - Atkinson, Jordan (2000) **Self-citation (Atkinson Jordan)** (Correct)

....2 and the Forest Research Group (FRG) at Sun Labs, 3 in October 1995. **We** initially envisaged PADRG building orthogonally persistent platforms (OPPs) and the FRG using them to build JP. **The design and first prototype of PJama (we ll call it PJama 0:0) was completed by the PADRG by July 1996 [20, 16].** It was based on the JDK1.0 and used the architecture summarised in Figure 4.1.a. Durability and atomicity depended on Recoverable Virtual Memory [188] there was a buffer pool into which pages were brought and object addresses on disk (PIDs) were byte offsets from the start of the file. **There** ....

....Pool Standard GC Heap Cache Object Disk etc. **Disk** etc. **Combined** Object Cache and GCheap Store Adapter Sphere Java Application Modified JVM Figure 4. **1: Persistence Architectures used for Versions of PJama The original proposals for PJama included a flexible transaction mechanism [20, 64](10) The first experimental investigation of this was developed in summer 1998 by running JavaInJava (JIJ) 205] on PJama 0:1 , as PJama t .** This gave full orthogonality but paid a high penalty in performance. **Work** is now underway to incorporate the flexible transaction model into the ....

[Article contains additional citation context not shown here]

M.P. Atkinson, M.J. Jordan, L. Daynes, and S. Spence. *Design Issues for Persistent Java: A Typesafe*, Object-oriented, Orthogonally Persistent System. In Connor and Nettles [51], pages 33--47.

Efficient support for customizing concurrency control.. - Daynes, Atkinson.. (1996) (2 citations) **Self-citation (Atkinson Dayn)** (Correct)

....accesses. **Keywords**: Persistent Java, Customizable Locking Mechanisms, Implementation, Optimisation for memoryresidence of objects. **1 Introduction We report on the issues raised when designing the addition and the implementation of a customisable locking mechanism for Persistent Java (PJava) [3], a type safe, object oriented, orthogonally persistent system based on the language Java [21] PJava is an alternative platform for the Java language with provision of completely orthogonal persistence [4] for data, meta data (classes) and code (methods) Orthogonal persistence was conceived in ....**

....don t meet our needs either. 1.2 Transaction in PJava This section briefly introduces the features of PJava relevant to this paper. **We** assume the reader is familiar with the main features of the Java language. **The reader is referred to [11] for a full description of the Java language and to [3, 2] for more thorough descriptions of PJava.** PJava s design presumes two categories of application programmer behaviour: 1. **defining** new transaction models, and 2. **using** predefined transaction models including those predefined in PJava. **Those** defining new transaction models specialise an abstract ....

[Article contains additional citation context not shown here]

M.P. Atkinson, M.J. Jordan, L. Dayn`es, and S. Spence. *Design issues for persistent Java: A typesafe*, object-oriented, orthogonally persistent system, May 1996. In the pre-proceedings of the 7th International Workshop on Persistent Object System (POS 7).

Main-Memory Management to support Orthogonal Persistence for Java - Laurent Dayn **Self-citation (Atkinson Dayn)** (Correct)

....information about the state and operations of the abstract machine. **We** present details of the mechanisms developed as we are not aware of them being described elsewhere and we believe they may be useful to other implementors of persistent languages and OODB bindings. **As stated previously [6, 4], our overall strategy was to devise an architecture which made objects that have been faulted in, and their house keeping data structures, look very similar to the representations used by the unmodified JVM to avoid making extensive changes to the JVM.** Like many persistent object systems, PJama 0 ....

....in enabling code, available either as source Java or compiled class files, prepared for other environments, to be re used in the PJama context without any transformation. **An extensive discussion of orthogonal persistence is given in [7] and its application to the language Java is described in [4, 6].** Using PJama, the bulk of an application suite consists of Java classes and methods written exactly as they would have been written if the application was transient, using only main memory for its data structures. **Each** program, typically the.class containing the main method, requires a few lines ....

M.P. Atkinson, M.J. Jordan, L. Dayn`es, and S. Spence. *Design issues for persistent Java: A typesafe*, object-oriented, orthogonally persistent system, May 1996. In the pre-proceedings of the 7th International Workshop on Persistent Object System (POS 7).

And Finally - John Zigman And (Correct)

No context found.

ATKINSON,M.P.,JORDAN, M. J., DAYN ES, L., AND SPENCE, S. 1996. *Design issues for Persistent Java: A type-safe*, object-oriented, orthogonally persistent system. In R. CONNOR AND S. NETTLES Eds., Seventh International Workshop on Persistent Object Systems (Cape May, NJ, U.S.A., May 1996), pp.

Virtual Virtual Machines - Bertil Folliot Ian (1997) (Correct)

No context found.

Atkinson, M.P. et al, *Design issues for persistent Java: A type-safe*, object-oriented, orthogonally persistent system, in Proc. 7th International Workshop on Persistent Object Systems (POS 7).

Persistence in CORBA - Tuma (1997) (Correct)

No context found.

Atk96A Atkinson, M. P. & Daynes, L. & Jordan, M. & Spence, S.: *Design issues for persistent Java: A type-safe*, object-oriented, orthogonally persistent system, in proceedings of The 7th International Workshop on Persistent Object Systems, Cape May, NJ USA, Morgan Kaufmann, 1996.

Analysing a Simple Disk Garbage Collector - Printezis (1996) (3 citations) (Correct)

No context found.

M. P. Atkinson, M. Jordan, L. Daynès, and S. Spence. *Design Issues for Persistent Java: a TypeSafe*, Object-Oriented, Orthogonally Persistent System, University of Glasgow, Scotland, February 1996. In Proceedings of POS'7.

First 50 documents